NAME (from your UF ID): _____ UF ID#: _____

(Please **PRINT**)

---------- CEN 4072/6070 Software Testing & Verification ----------

Exam 2 – Spring 2017

You have 50 minutes to work on this exam. It is a "closed-book/closed-notes" test. Pay attention to point values, since you may not have time to work all 11 problems.

PRINT your name and UF ID# above NOW and sign the pledge at the bottom of this page, if appropriate, when you are finished.

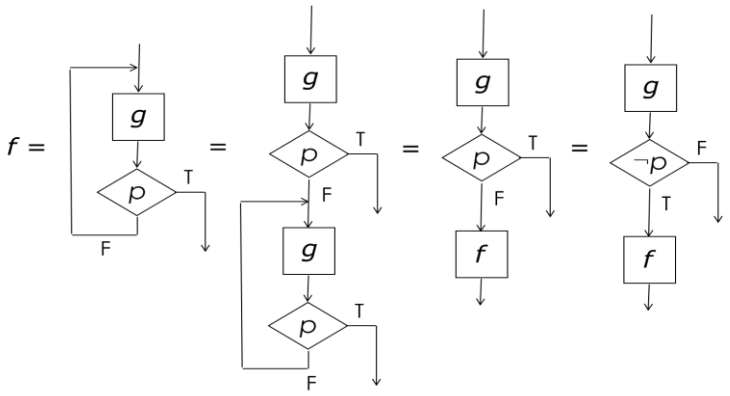
PLEASE PRINT **– do NOT write *cursively* –** ANSWERS IN THE SPACE PROVIDED ONLY – **NOT IN THE MARGINS** – PREFERABLY USING A BALLPOINT PEN TO INCREASE LEGIBILITY. Good luck!

On my honor, I have neither given nor received unauthorized aid on this exam and I pledge not to divulge information regarding its contents to those who have not yet taken it.

_____

SIGNATURE

1. (3 pts.) The diagram below was used in class to illustrate an important concept/result related to functional verification.



Which one of the following concepts/results was derived in connection with the control flow relationships illustrated? (Circle ONE only.)

a. the Rule of Inference for proving {P} *repeat g until p* {Q}

b. Subgoal Induction

c. the weakest possible *f*-adequate loop invariant for [*repeat g until p*]

d. the Axiom of Replacement

e. the weakest pre-condition of *repeat g until p* with respect to post-condition Q

f. the correctness conditions for *f* = [*repeat g until p*]

g. the functional relationship between repeat_until and while-do constructs

2. (3 pts.) Consider the following assertion of weak correctness. (You may assume that N represents the number of elements in integer array List[1:N] and that the value of integer variable Index is always ≤ N.)

```
{N≥1}
      Index := 0
      repeat
         Index := Index+1
         Found := (Key=List[Index])
      until (Found OR Index=N)
{(Found ∧ Key=List[Index]) ∨ (¬Found ∧ ∀ (1≤i≤N) • key≠List[i])
```

Which one of the following expressions could be used as a Q-adequate Invariant to prove this assertion using the standard **Repeat_Until** ROI? (Circle ONE only.)

i. (Found ∧ Key=List[Index] ∧ ∀ (1≤i<Index) • Key≠List[i]) V (¬Found)

ii. (Found ∧ Key=List[Index]) V (¬Found ∧ ∀ (Index<i≤ N) • Key≠List[i])

iii. (Found ∧ Key=List[Index] ∧ ∀ (1≤i<Index) • Key≠List[i])

iv. [(Found ∧ Key=List[Index]) V (¬Found ∧ ∀ (i ∈ [1,Index) U (Index,N]) • Key≠List[i])] ∧ iorder (where iorder = ∀ 1≤i<N • List[i]≥List[i+1])

v. Key=List[Index] V (¬Found)

vi. (Found ∧ Key=List[Index]) V (¬Found ∧ ∀ (1≤i≤Index) • Key≠List[i])

vii. true

3. a. (4 pts.) In Quiz 3, you are asked to indicate whether the following hypothesized ROI is valid or invalid:

$$\frac{\{true\}\ S\ \{I\}\ \text{strongly},\ (I \wedge b) \Rightarrow Q}{\{P\}\ \text{repeat S until b}\ \{Q\}\ \text{strongly}}\ ?$$

The correct answer is **invalid.** Which one of the following constitutes a counterexample that *proves* this? (Circle ONE only.)

i. Suppose P is x=5, S is (x := x+1), b is x=0, Q is x=0, and I is y=17.

ii. Suppose P is x=5, S is (x := x+1; y := 17), b is x=0, Q is (y=17 Λ x=0), and I is y=17.

iii. Suppose P is x=-5, S is (x := x+1; y := 17), b is x=0, Q is (y=17 Λ x=0), and I is y=17.

iv. Suppose P is (x=-5 Λ y=17), S is (x := x+1), b is x=0, Q is (y=17 Λ x=0), and I is y=17.

v. (none of the above)

b. (4 pts.) In Quiz 3, you are asked to indicate whether the following hypothesized ROI is valid or invalid:

$$\frac{(\neg b) \Rightarrow (P \wedge Q)}{\{P\}\ \text{while b do S}\ \{Q\}}\ ?$$

The correct answer is **valid.** Which one of the following assertions could be proven by using this ROI? (Circle ONE only.)

i. {y=17} while (x>0 V y<>17) do x := x-1 {x≤0}

ii. {y=17} while (x>0) do x := x-1 {x≤0}

iii. {x≤0} while (x>=0) do (x := x-1; y := 17) {y=17 ∧ x≤0}

iv. {y=17} while (x>0 ∧ y<>17) do x := x-1 {x≤0}

v. (none of the above)

c. (4 pts.) In Quiz 3, you are asked to indicate whether the following hypothesized ROI is valid or invalid:

$$\frac{P \Rightarrow (b \wedge Q)}{\{P\}\ \text{repeat S until b}\ \{Q\}}\ ?$$

The correct answer is **invalid.** Which one of the following constitutes a counterexample that *proves* this? (Circle ONE only.)

i. Suppose P is x=0, S is (x := 0), b is x<=0, and Q is x≥0.

ii. Suppose P is x=0, S is (x := x-17), b is x=0, and Q is x≥0.

iii. Suppose P is x≥0, S is (x := x-17), b is x=0, and Q is x≥0.

iv. Suppose P is x=0, S is (x := x-17), b is x<=0, and Q is x≥0.

v. (none of the above)

4. (14 pts.) From Problem Set 6, it was determined that H_1, H_2, H_3, and H_k in the *open-form* expression of the *weakest pre-condition* (wp) of program R:

```
Repeat
  x := x+1;
  y := y-1
Until y=0
```

with respect to post-condition (x=17) are:

H_1: $y=1 \wedge x=16$ $\quad\quad$ H_3: $y=3 \wedge x=14$

H_2: $y=2 \wedge x=15$ $\quad\quad$ H_k: $y=k \wedge x=(17-k)$

where x and y are integers.

Answer questions (a) through (g) below by identifying the **SINGLE MOST APPROPRIATE EXPRESSION OR VALUE AMONG THE FOLLOWING.** Expressions/values may apply to none, one, or more than one question. (Note: assume that y' represents the *initial* value of variable y.)

A. $(y=0 \wedge x=5+y') \vee$ "undefined"
B. $H_1 \vee H_2 \vee H_3 \vee ... \vee H_k \vee ...$
C. $y \geq 0 \wedge x=17-y$
D. $y>0 \wedge x=17-k$
E. true
F. $(x=17-y) \vee (y<0)$
G. $x=7-y$
H. $[y'>0 \Rightarrow (y=0 \wedge x=5+y')] \wedge (y' \leq 0 \Rightarrow$ "undefined"$)$
I. $(x=7-y) \wedge (y \geq 0)$
J. $k>0 \wedge x=17-k$
K. $(y=0 \wedge x=5+y')$
M. $y>0 \wedge x=17-y$
O. $x=7 \wedge y=0$
P. "undefined"
R. $(x=17-y) \vee (y \leq 0)$

_____ a. What is wp(R, x=17) in ***closed form?***

_____ b. What is the *weakest Q-adequate* loop Invariant for R with respect to post-condition (x=17) that guarantees termination?

_____ c. What is wlp(R, x=17) in *closed form*?

_____ d. What is the *weakest Q-adequate* loop Invariant for R with respect to post-condition (x=17) that does **NOT** guarantee termination?

_____ e. What is sp(R, $x=5 \wedge y=2$)?

_____ f. What is sp(R, $x=5 \wedge y=-2$)?

_____ g. What is sp(R, x=5)?

5. (8 pts.) Give the four Subgoal Induction correctness conditions for proving

$$f = [K] = [h;\ \text{while } p \text{ do } g \text{ end\_while};\ t]$$

where v = [while p do g end_while; t].

6. (5 pts.) Consider the following program *P*:

```
y := 3;
if y>0 then
   x := x+y
else
   x := x-y
end_if_else
y := x*y
```

Which one of the following is [*P*]? (Circle ONE only.)

a. (x>0 -> x,y := 4x,3 | x≤0 -> x,y := x-3y,3)

b. (xy>0 -> x,y := 4x,3 | xy≤0 -> x,y := x-3y,3)

c. (xy>0 -> x,y := x+xy,3 | true -> x,y := xy-x,3)

d. (xy>0 -> x,y := x+xy,3 | xy<0 -> x,y := x-xy,3 | true -> x,y := x,3)

e. (y>0 -> x,y := $x+y,xy+y^2$) | y≤0 -> x,y := $x-y,xy-y^2$)

f. (none of the above)

7. (4 pts.) Identify specific initial (i.e., "input") variable values that prove the assertion:

{z<0} S {y=z+1}

is false given that program S is:

```
while z <> -5 do
   z := z+1;
   y := z+1
end_while
```

Briefly explain how/why these initial values demonstrate that the assertion is false.

8. Suppose you are asked to prove that the program:

```
K:  repeat
        x := x-1;
        y := y*2
    until x=0
```

computes the function:

$$f = (x>0 \rightarrow x,y := 0,\ y2^x)$$

by showing that each of the following correctness conditions hold:

(1) term(f,K),

(2) $(p \circ g) \Rightarrow (f=g)$ and

(3) $\neg(p \circ g) \Rightarrow (f=f \circ g)$,

a. (2 pts.) Express "*g*" as a single concurrent function in terms of variables x and y?

b. (1 pt.) What is "*p*" from correctness conditions (2) and (3) above? (Give the actual expression that *p represents*.)

p: _____

c. (3 pts.) Assume that you wish to use the Method of Well-Founded Sets to prove term(*f,K*).

i. Give a simple "measure" (and ONLY a measure) that could be used when applying this method with *K.*

measure: _____

ii. For what range of possible initial values of the measure would the proof be relevant?

iii. By what value would the measure be "bounded" – i.e. what would be the last possible value of the measure given the range of initial values identified in part (ii) above?

d. (9 pts.) Prove that correctness condition (3) holds. **Show and explicitly justify all steps and cases as illustrated in class, supplied problem solutions, etc.**

(Provide your proof on the next page.)

8. d. (cont'd)

K: repeat
x := x-1;
y := y∗2
until x=0

$f = (x>0 \rightarrow x,y := 0, y2^x)$

Proof that $\neg(p \circ g) \Rightarrow (f = f \circ g)$:

9. a. (5 pts.) Consider the intended function f = (x≥0 -> x,y := 0,y+2x). Which one of the following is the **weakest *f*-adequate invariant *q* over the *D*(*f*) for while loops computing *f*** *?* (Circle ONE only.)

i. $y+2x_0 = y_0+2x \wedge x\geq 0$

ii. $y+2x = y_0+2x_0 \wedge y\geq 0$

iii. (x≥0 -> x,y := x-1,y+2)

iv. $y+x = y_0+x_0 \wedge x\geq 0$

v. $0=0 \wedge y=y_0+2x_0 \wedge x\geq 0$

vi. $y=y_0+2(x_0-x) \wedge x\geq 0$

b. (4 pts.) Consider the program *T*: **while *p* do *g* end_while** where *p* is x<>0 and *g* = (x,y := x-1,y+2) which computes *f.* Give the sequence of states X_0, X_1, ..., X_n produced by each iteration of *T* given that the initial state, $X_0 = (x_0,y_0) = (3,3)$ and X_n is the final state.

c. (12 pts.) Give the values for each of the following expressions. (Assuming $X_0 = (3,3)$.)

$p(X_0)$: _____ $g(X_0)$: _____ $f(X_2)$: _____ $q(X_2)$: _____

$p(X_2)$: _____ $g(X_2)$: _____ $f(X_n)$: _____ $q(2,7)$: _____

$p(X_n)$: _____ $g(X_n)$: _____ $f(g(X_n))$: _____ $q(g(X_n))$: _____

d. (3 pts.) For what initial (input) state $X_0 = (x_0,y_0)$, if any, would $X_2 = (2,7)$ "agree with *q*"?

e. (4 pts.) Suppose $X_0 = (x_0,y_0) = (177,-17)$. (Note: To compensate for random guessing, your score in points for part (e) will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of an answer, consider skipping the question.)

i. Does *T* (from part "b", above) produce the intermediate state (88,163) from this input? (Circle One only.)

yes no

ii. Is there *any other while loop which computes f* that does so? (Circle One only.)

yes no

10. (10 pts.) The following statements relate to King, et al., "Is Proof More Cost Effective than Testing?" Indicate whether each is true or false. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the problem.

a. The application described in the paper is a safety critical missile abort system developed for the UK Ministry of Defense (MOD) that is used by range safety officers during missile test firings.	true	false
b. The "SPARK" notation was used for documenting the system specification and part of the design, and a subset of Ada was used for coding.	true	false
c. The authors found that a good (informal) understanding of the meaning of imperative programming constructs, together with some familiarity with relevant proof concepts, such as loop invariants, were necessary to conduct the formal code proofs.	true	false
d. Code proofs appeared to be somewhat less efficient than unit testing, probably because substantial amounts of unit testing were completed before the bulk of code proof started.	true	false
e. Z proofs, however, appeared to be substantially more efficient at finding faults than the most efficient testing phase.	true	false

11. (10 pts.) The following statements relate to Linger, "Cleanroom Software Engineering for Zero-Defect Software." Indicate whether each is true or false. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the problem.

a. The paper's title comes from the author's claim that knowing for certain that a software product has zero defects becomes possible when correctness verification is an integral part of the development process.	true	false
b. The fifteen successful projects described in the paper that utilized Cleanroom SE consisted of systems developed by IBM, NASA, the University of Tennessee, and Martin Marietta.	true	false
c. Unlike coverage testing, the statistical testing employed in Cleanroom SE is biased with respect to finding errors in failure rate order on average.	true	false
d. A sample population of user executions based on expected frequency is used for low-probability, high-consequence functions.	true	false
e. Code execution is not permitted by the development team; team correctness verification takes the place of unit testing and debugging.	true	false